UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/643,628	08/18/2003	Wei Li	50277-2250	4451

42425 7590 04/27/2009

HICKMAN PALERMO TRUONG & BECKER/ORACLE
2055 GATEWAY PLACE
SUITE 550
SAN JOSE, CA 95110-1083

EXAMINER	
SAEED, USMAAN	

ART UNIT	PAPER NUMBER
2166	

MAIL DATE	DELIVERY MODE
04/27/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 10/643,628
Filing Date: August 18, 2003
Appellant(s): LI ET AL.

Daniel D. Ledesma
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 01/12/2009 appealing from the Office action mailed 08/11/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The following are the related appeals, interferences, and judicial proceedings known to the examiner which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal:

10/643,629

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,324,533	Agrawal	11-2001
2002/0087561	Ching Chen	7-2002
6,138,117	Bayardo	10-2000
2002/0059191	Tamura	5-2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation

under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-2, 4-7, 12-15, 17-20, and 25-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Agrawal et al. (Agrawal** hereinafter) (U.S. Patent No. 6,324,533) in view of **Ching Chen et al**. **(Chen** hereinafter) (U.S PG Pub No. 2002/0087561).

With respect to claim 1, **Agrawal** teaches **a method for performing a frequent itemset operation, the method comprising the steps of:**

"within a database server that supports a particular database language, parsing a database statement to detect within the database statement a construct that extends the particular database language" as an object of the present invention is to provide a method for mining data relationships from the integrated mining system in the form of queries to SQL engines, and with k-way join, three-way join, subqueries, and group-by operations for counting the itemset support **(Agrawal** Col 2, Lines 27-31). A method for mining data relationships from the integrated mining system in the form of queries to SQL engines enhanced with object-relational extensions (SQL-OR), such as user-defined functions (UDFs) and table functions **(Agrawal** Col 2, Lines 33-36).

"wherein the construct identifies a function that counts and return frequent itemsets" as the group-by query preferably includes the steps of counting the number of transactions that contain each item and selecting the items that have a support above a user-specified threshold in determining the frequent itemsets (**Agrawal** Col 2, Lines 53-56).

"wherein the function identifies said frequent item sets obtained by the statement" as (**Agrawal** Figure 3 and 9).

"performing said frequent itemset operation as part of execution of the database statement to produce results" as the mining operation is expressed in some extension of SQL or a graphical language, which are input to preprocessor 21. This preprocessor generates appropriate SQL translations for the mining operation. For example, these SQL translations may be those that are executed by a SQL-92 relational engine 22. It is assumed that blobs, user-defined functions, and table functions are available in the object-relational engine. The mining results might be output to a depository 24 (**Agrawal** Col 6, Lines 26-42 and Figure 3).

"storing the results in a computer-readable medium" as figure 1 reference numeral 9 (**Agrawal** Figure 1).

Agrawal teaches the elements of claim 1 as noted above but does not explicitly teaches **"a cursor as input and wherein the cursor is used by the function to access values from rows that are returned from a select statement."**

However, **Chen** teaches **"a cursor as input"** as control begins at block 200 with the executive 6 receiving an OPEN command for a static cursor scroll. The DECLARE

statement for the static scrollable cursor would have been previously processed. The executive 6 then calls (at block 202) the parser compiler 8 and optimizer 10 to parse and optimize the OPEN statement. After the OPEN statement is parsed and optimized, the executive 6 calls (at block 204) the structure generator 12 to construct an INSERT command from the SELECT statement in the previously compiled and executed DECLARE statement to populate the rows of the result table 50 with the qualifying rows of the base table 60 (**Chen** Paragraph 0051).

"wherein the cursor is used by the function to access values from rows that are returned from a select statement" as the declaration of the cursor would provide a SELECT statement specifying columns of the database table 60 and a WHERE clause including one or more predicates to qualify rows of the database table 60. The data manager 16 would return to the cursor the selected columns in the select list from rows that satisfy the WHERE statement (**Chen** Paragraph 0032).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Chen's** teachings would have allowed **Agrawal** to provide high concurrency for the rows in the base table when cursors are used by obtaining a lock on the rows in the base table for the duration of the cursor operation.

With respect to claim 2, **Agrawal** teaches **"the method of claim 1, wherein the database statement is expressed in a particular database language, and wherein the particular database language is SQL"** as a method for mining data relationships

from the integrated mining system in the form of queries to SQL engines enhanced with object-relational extensions (SQL-OR), such as user-defined functions (UDFs) and table functions (**Agrawal** Col 2, Lines 33-36).

With respect to claim 4, **Agrawal** teaches **the method of claim 1 wherein:**

"the database statement includes a first indication of a first input format" as the data table is first transformed into a vertical format by creating for each item a BLOB containing all tids that contain that item (Tid-list creation phase) and then count the support of itemsets by merging together these tid-lists (support counting phase) (**Agrawal** Col 12, Lines 43-47).

"the frequent itemset operation operates on input that conforms to said first input format" as a table function Gather is used for creating the Tid-lists. This is the same as the Gather function in GatherJoin except here, the tid-list is created for each frequent item. The data table T is scanned in the (item, tid) order and passed to the function Gather. The function collects the tids of all tuples of T with the same item in memory and outputs a (item, tid-list) tuple for items that meet the minimum support criterion. The tid-lists are represented as BLOBs and stored in a new TidTable with attributes (item, tid-list) (**Agrawal** Col 12, Lines 48-56).

"the method further comprises the steps of: parsing a second database statement to detect within the second database statement the construct that extends a database language" as a method for mining data in an integrated database

and data-mining system. Start with step 30, a group-by query is performed on the data transactions to generate a set of frequent 1-itemsets. One-itemsets are those having exactly one item each, while an itemset is frequent if the number of transactions containing it is at least at a specified number. At step 31, frequent 2-itemsets are determined from the frequent 1-itemsets and the transaction table. A candidate set of (n+2)-itemsets is next generated in step 32 from the frequent (n+1)-itemsets, where n=1. At step 33, frequent (n+2)-itemsets are generated from the candidate set of (n+2)-itemsets and the transaction table using a query (**Agrawal** Col 6, Lines 43-55). A first query is being performed to generate 1-itemsets, and (n+2) itemsets are being generated using another query. **"wherein the second database statement includes a second indication of a second input format that is different from said first input format"** as a horizontal format where each tid is followed by a collection of all its items (**Agrawal** Col 10, Lines 37-38).

"in response to detection of said construct in said second database statement, the database server performing a second frequent itemset operation as part of execution of the second database statement" as the mining operation is expressed in some extension of SQL or a graphical language, which are input to preprocessor 21. This preprocessor generates appropriate SQL translations for the mining operation. For example, these SQL translations may be those that are executed by a SQL-92 relational engine 22. It is assumed that blobs, user-defined functions, and table functions are available in the object-relational engine. The mining results might be output to a depository 24 (**Agrawal** Col 6, Lines 26-42). **"wherein the second frequent**

itemset operation operates on input that conforms to said second format" as K-way Join approach where the k-way self join of T is replaced with the table functions Gather and Comb-K. It is possible to merge these functions together as a single table function GatherComb-K. The Gather function is not required when the data is already in a horizontal format where each tid is followed by a collection of all its items (**Agrawal** Col 10, Lines 33-38).

With respect to claim 5, **Agrawal** teaches **"the method of claim 4 wherein the first indication is identification of a first table function"** as a table function Gather is used for creating the Tid-lists. This is the same as the Gather function in GatherJoin except here, the tid-list is created for each frequent item. The data table T is scanned in the (item, tid) order and passed to the function Gather. The function collects the tids of all tuples of T with the same item in memory and outputs a (item, tid-list) tuple for items that meet the minimum support criterion (**Agrawal** Col 12, Lines 48-56). **"and the second indication is identification of a second table function"** as the output of Gather is passed to another table function Comb-K which returns all k-item combinations formed out of the items of a transaction (**Agrawal** Col 10, Lines 24-27).

With respect to claim 6, **Agrawal** teaches **"the method of claim 1 wherein the frequent itemset operation uses, as input, a row source that is generated during execution of other operations specified in said database statement"** as output is a

collection of rules of varying length. The maximum length of these rules is much smaller than the number of items and is rarely more than a dozen. Therefore, a rule is represented as a tuple in a fixed-width table where the extra column values are set to NULL to accommodate rules involving smaller itemsets. The schema of a rule is (item.sub.1, . . . , item.sub.k, len, rulem, confidence, support) where k is the size of the largest frequent itemset (**Agrawal** Col 5, Lines 65-67 & Col 6, Lines 1-6). A table function, GenRules, is used to generate all possible rules from a frequent itemset. The input to the function is a frequent itemset. For each itemset, it outputs tuples corresponding to rules with all non-empty proper subsets of the itemset in the consequent. The table function outputs tuples with k+3 attributes, T_item.sub.1, . . . , T_item.sub.k, T_support, T_ten, T_rulem (**Agrawal** Col 8, Lines 7-13). From first operation a row/tuple is being obtained, which is then being used as an input.

With respect to claim 7, **Agrawal** teaches **"the method of claim 1 wherein the frequent itemset operation produces, as output, a row source that is used as input for other operations specified in said database statement"** as output is a collection of rules of varying length. The maximum length of these rules is much smaller than the number of items and is rarely more than a dozen. Therefore, a rule is represented as a tuple in a fixed-width table where the extra column values are set to NULL to accommodate rules involving smaller itemsets. The schema of a rule is (item.sub.1, . . . , item.sub.k, len, rulem, confidence, support) where k is the size of the largest frequent itemset (**Agrawal** Col 5, Lines 65-67 & Col 6, Lines 1-6). A table

function, GenRules, is used to generate all possible rules from a frequent itemset. The input to the function is a frequent itemset. For each itemset, it outputs tuples corresponding to rules with all non-empty proper subsets of the itemset in the consequent. The table function outputs tuples with k+3 attributes, T_item.sub.1, . . . , T_item.sub.k, T_support, T_ten, T_rulem (**Agrawal** Col 8, Lines 7-13). From first operation a row/tuple is being obtained as an output, which is then being used as an input.

With respect to claim 12, **Agrawal** teaches **"the method of claim 1 wherein the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results identify frequent itemsets, and for each of the frequent itemsets, a count of how many item groups included the frequent itemset"** as a set of frequent 1-itemsets is generated using a group-by query on data transactions. From these frequent 1-itemsets and the transactions, frequent 2-itemsets are determined. A candidate set of (n+2)-itemsets are generated from the frequent 2-itemsets, where n=1. Frequent (n+2)-itemsets are determined from candidate set and the transaction table using a query operation (**Agrawal** Abstract).

With respect to claim 13, **Agrawal** teaches **"the method of claim 1 wherein the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results identify frequent itemsets, and for each of the**

frequent itemsets, a count of how items are in the frequent itemset" as a set of frequent 1-itemsets is generated using a group-by query on data transactions (**Agrawal** Abstract). The support counting phase, conceptually for each itemset in C.sub.k the tid-lists of all k items are collected and the number of tids in the intersection of these k lists is counted using a user defined function (UDF) (**Agrawal** Col 12, Lines 56-59).

Claims 14-15, 17-20 and 25-26 are essentially the same as claims 1-2, 4-7, and 12-13, except they set forth the claimed invention as a computer readable media carrying instructions and are rejected for the same reasons as applied hereinabove.

With respect to claims 27 and 28, **Agrawal** teaches **"wherein the construct is a table function"** as (Agrawal Col 10, Lines 32-36).

Claims 3, 9-11, 16, and 22-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Agrawal et al.** (U.S. Patent No. 6,324,533) in view of **Ching Chen et al**. (U.S PG Pub No. 2002/0087561) as applied to claim 1-2, 4-7, 12-15, 17-20, and 25-26 above further in view of **Roberto Javier Bayardo**. (**Bayardo** hereinafter) (U.S Patent No. 6,138,117).

With respect to claim 3, **Agrawal** teaches **"the method of claim 1, wherein the construct is a table function"** as a method for mining data relationships from the

integrated mining system in the form of queries to SQL engines enhanced with object-relational extensions (SQL-OR), such as user-defined functions (UDFs) and table functions (**Agrawal** Col 2, Lines 33-36).

"wherein the database statement specifies frequency criteria and additional criteria, wherein said frequency criteria specifies at least one criterion that relates to how frequently combination of items appear together" as to find all combinations of items whose support is greater than minimum support. Call those combinations frequency itemsets (**Agrawal** Col 5, Lines 20-23).

"wherein the results include frequent itemsets that satisfy both said frequency criteria and said additional criteria, and wherein the results do not include frequent itemsets that satisfy said frequency criteria but do not satisfy said additional criteria" as the frequent (n+2)-itemsets are determined using cascaded subqueries by: a) selecting distinct first items in the candidate itemsets using a subquery (**Agrawal** Col 3, Lines 2-4). Using the results of the last subqueries to determine which of the (n+2)-itemsets are frequent. In generating rules from the union of the frequent itemsets, all items from the frequent itemsets are first put into a table F. A set of candidate rules is created from the table Fusing a table function. These candidate rules are joined with the table F, and filtered to remove those that do not meet a confidence criteria (**Agrawal** Col 3, Lines 9-16).

F consists of k+2 attributes (item.sub.1, . . . , item.sub.k, support, len), where k is the size of the largest frequent itemset and len is the length of the itemset (**Agrawal** Col 8, Lines 4-6). Sequence of operations can be implemented as a single SQL query for any

k, as shown in FIG. 12. Therefore the query specifies both the frequency criteria and the additional criteria k, which is the size of an itemset.

Agrawal teaches the elements of claim 3 as noted above but does not explicitly disclose, **"wherein said additional criteria do not specify any criterion that related to how frequently combinations of items appear together" and "the additional criteria specify at least one of (a) minimum length (b) maximum length (c) a set of one or more included items or (d) a set of one or more excluded items."**

However, **Bayardo** discloses, **"wherein said additional criteria do not specify any criterion that relates to how frequently combinations of items appear together"** as Max-Miner usually performs less database passes than this bound in practice when the longest frequent itemsets are more than 10 in length (**Bayardo** Col 9, Lines 57-60). Examiner interprets the length of 10 as additional criteria.

"the additional criteria specify at least one of (a) minimum length (b) maximum length (c) a set of one or more included items or (d) a set of one or more excluded items" as Max-Miner usually performs less database passes than this bound in practice when the longest frequent itemsets are more than 10 in length (**Bayardo** Col 9, Lines 57-60). Examiner interprets the length of 10 as the minimum length.

The most part, frequent-pattern mining methods have been developed to operate on databases in which the longest frequent patterns are relatively short, e.g., those with less than 10 items (**Bayardo** Col 1, Lines 22-26). Examiner interprets the length of 10 as the maximum length.

A method for identifying patterns from a database of records including the steps of: (1) generating an initial set C of candidates where each candidate c includes two distinct sets of items: c.head and c.tail (**Bayardo** Col 3, Lines 42-45).

It is still another object of the present invention to quickly identify those patterns that are both frequent and maximal so that the set of maximal frequent patterns represents the set of all frequent patterns (**Bayardo** Col 3, Lines 32-35 and Lines 40-56).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Bayardo's** teachings would have allowed **Agrawal and Chen** to provide an efficient method for extracting relatively long frequent patterns from a database of transaction records where each record includes several data items.

With respect to claim 9, **Agrawal** teaches **"the method of claim 1 wherein: the additional criteria specify a minimum length; and the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results exclude all item sets that include fewer items than the minimum length specified by the additional criteria"** as combinations of items whose support is greater than minimum support. Call those combinations frequent itemsets (**Agrawal** Col 5, Lines 21-23). The function collects the tids of all tuples of T with the same item in memory and outputs a (item, tid-list) tuple for items that meet the minimum support criterion (**Agrawal** Col 12, Lines 52-55).

Agrawal further teaches the function collects the tids of all tuples of T with the same item in memory and outputs a (item, tid-list) tuple for items that meet the minimum support criterion. The tid-lists are represented as BLOBs and stored in a new TidTable with attributes (item, tid-list) (**Agrawal** Col 11, Lines 49-56).

Agrawal teaches the elements of claim 9 as noted above but does not explicitly teaches **"a minimum length."**

However, **Bayardo** teaches **"a minimum length"** as Max-Miner usually performs less database passes than this bound in practice when the longest frequent itemsets are more than 10 in length (**Bayardo** Col 9, Lines 57-60). Examiner interprets the length of 10 as the minimum length.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Bayardo's** teachings would have allowed **Agrawal and Chen** to provide an efficient method for extracting relatively long frequent patterns from a database of transaction records where each record includes several data items.

With respect to claim 10, **Agrawal** teaches **"the method of claim 1 wherein: the additional criteria specify a maximum length; and the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results exclude all item sets that include more items than the maximum length specified by the additional criteria"** as F consists of k+2 attributes (item.sub.1,

. . . , item.sub.k, support, len), where k is the size of the largest frequent itemset and len is the length of the itemset (**Agrawal** Col 8, Lines 4-6).

Agrawal further teaches in particular, it is not practical to assume that all items in a transaction appear as different columns of a single tuple because often the number of items per transaction can be more than the maximum number of columns that the database supports. For instance, for one of our real-life datasets the maximum number of items per transaction is 872 and for another it is 700 (**Agrawal** Col 5, Lines 56-60).

Agrawal teaches the elements of claim 10 as noted above but does not explicitly teaches **"a maximum length."**

However, **Bayardo** discloses **"a maximum length"** as the most part, frequent-pattern mining methods have been developed to operate on databases in which the longest frequent patterns are relatively short, e.g., those with less than 10 items (**Bayardo** Col 1, Lines 22-26). Examiner interprets the length of 10 as the maximum length.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Bayardo's** teachings would have allowed **Agrawal and Chen** to provide an efficient method for extracting relatively long frequent patterns from a database of transaction records where each record includes several data items.

With respect to claim 11, **Agrawal** teaches **"the method of claim 1 wherein: the additional criteria specify a set of one or more included items; and the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results exclude all itemsets that do not include all items in said set of one or more included items"** as the frequent (n+2)-itemsets are determined using cascaded subqueries by: a) selecting distinct first items in the candidate itemsets using a subquery. In generating rules from the union of the frequent itemsets, all items from the frequent itemsets are first put into a table F. These candidate rules are joined with the table F, and filtered to remove those that do not meet a confidence criteria (**Agrawal** Col 3, Lines 2-16).

Agrawal teaches the elements of claim 11 as noted above but does not explicitly teaches **"one or more included items."**

However, **Bayardo** discloses **"one or more included items"** as a method for identifying patterns from a database of records including the steps of: (1) generating an initial set C of candidates where each candidate c includes two distinct sets of items: c.head and c.tail (**Bayardo** Col 3, Lines 42-45).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Bayardo's** teachings would have allowed **Agrawal and Chen** to provide an efficient method for extracting relatively long frequent patterns from a database of transaction records where each record includes several data items.

Claims 3 and 9-11 are essentially the same as claims 16 and 22-24, except they set forth the claimed invention as a computer readable media carrying instructions and are rejected for the same reasons as applied hereinabove.

Claims 29 and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Agrawal et al.** (U.S. Patent No. 6,324,533) in view of **Ching Chen et al**. (U.S PG Pub No. 2002/0087561) further in view of **Roberto Javier Bayardo** (U.S Patent No. 6,138,117), further in view of **Takayuki Tamura (Tamura** hereinafter) (U.S. Patent No. 20020059191).

With respect to claims 29 and 30, **Agrawal, Chen and Bayardo** do not teach **"the additional criteria specify a set of one or more excluded items and the step of performing the frequent itemset operation includes performing a frequent itemset operations whose results exclude all itemsets that include all items in said set of one more excluded items."**

However, **Tamura** discloses **"the additional criteria specify a set of one or more excluded items and the step of performing the frequent itemset operation includes performing a frequent itemset operations whose results exclude all itemsets that include all items in said set of one more excluded items"** as

An itemset including a combination of items, which is not in the frequent (k-1)-itemset, is excluded from the candidate k-itemset (**Tamura** Paragraph 0021).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teaching of the cited references because **Tamura's** teachings would have allowed **Agrawal, Chen, and Bayardo** to provide an efficient method by improving the performance of the data system by parallelization etc. the performance can be improved without changing the mining system.

(10) Response to Argument

A. §103(a) rejection of claims 1-2, 4-7, 12-15, 17-20, and 25-28 over Agrawal in view of Chen.

Appellant argues that Agrawal and Chen do not teach or suggest **"a function that counts and return frequent itemsets given a cursor as input to the function; wherein the cursor is used by the function to access values from rows that are returned from a select statement; wherein the function identifies said frequent itemsets based on said values from said rows returned by said select statement"** as required by independent claim 1.

In response to the preceding arguments examiner respectfully submits that **Agrawal** teaches **"a function that counts and return frequent itemsets" and "wherein the function identifies said frequent item sets obtained by the select**

statement" as the group-by query preferably includes the steps of counting the number of transactions that contain each item and selecting the items that have a support above a user-specified threshold in determining the frequent itemsets (**Agrawal** Col 2, Lines 53-56).

Agrawal further teaches the use of table functions described above. It generates all possible k-item combinations of items contained in a transaction, joins them with the candidate table C.sub.k, and <u>counts the support of the itemsets by grouping the join result</u>. Two table functions, Gather and Comb-K, are used. The data table T is scanned in the (tid, item) order and passed to the table function Gather. This table function collects all the items of a transaction (in other words, items of all tuples of T with the same tid) in memory and outputs a record for each transaction. Each such record consists of two attributes, the tid and item-list which is a collection of all its items in a VARCHAR or a BLOB. The output of Gather is passed to another table function Comb-K which returns all k-item combinations formed out of the items of a transaction. A record output by Comb-K has k attributes T_itm.sub.1, . . . , T_itm.sub.k, which can be directly used to probe into the C.sub.k table. An index is constructed on all the items of C.sub.k to make the probe efficient. FIG. 10 illustrates the SQL queries for the GatherJoin approach. This approach is analogous to the K-way Join approach where the k-way self join of T is replaced with the table functions Gather and Comb-K. It is possible to merge these functions together as <u>a single table function GatherComb-K</u>. The Gather function is not required when the data is already in a horizontal format

where each tid is followed by a collection of all its items. The pseudo-code below illustrate a typical implementation of GatherJoin approach for counting support.

insert into F.sub.k select item.sub.1, . . ., item.sub.k, count(*)

from C.sub.k,

(select t.sub.2.T_itm.sub.1, . . . , t.sub.2.T_itm.sub.k from T,

table (Gather(T.tid, T.item)) as t.sub.1,

table (Comb-K(t.sub.1.tid, t.sub.1.item-list)) as t.sub.2)

where t.sub.2.T_itm.sub.1 =C.sub.k.item.sub.1 and

t.sub.2.T_itm.sub.k =C.sub.k.item.sub.k

group by C.sub.k.item.sub.1, . . . , C.sub.k.item.sub.k

having count(*)>minsup

Note that for k=2, the 2-candidate set C.sub.2 is simply a join of F.sub.1 with itself. Accordingly, the pass 2 can be optimized by replacing the join with C.sub.2 by a join with F.sub.1 before the table function (see FIG. 10). That way, the table function gets only frequent items and generates significantly fewer 2-item combinations (**Agrawal** Col 10, Lines 13-56).

In these lines Examiner interprets single table function GatherComb-K as a function required by the applicant because this function is counting and generating frequent itemsets with 2-item combinations with k=2. Appellant argues that the Gather table function neither counts nor returns frequent itemsets but merely returns all k-item combinations formed out if items of a single transaction. Examiner respectfully submits that the above lines and figure 11 explicitly show that the GatherComb-K function is

<u>*being used for counting and returning all k item combinations for a transaction and this table function gets only frequent items with 2-item combinations for k=2 per transaction. Agrawal's Col 15, further teaches that the table function maintains a counter. Agrawal further teaches on col 11, lines 5-12, that the function updates the counts and outputs only the frequent itemsets after the last transaction.*</u>

<u>*Therefore, examiner interprets Agrawal's function containing a counter for counting and returning frequent itemsets after the last transaction as the function claimed by the applicant.*</u>

Agrawal does not teaches **"a cursor as input and wherein the cursor is used by the function to access values from rows that are returned from a select statement."**

However, **Chen** teaches **"a cursor as input"** as control begins at block 200 with the executive 6 receiving an OPEN command for a static cursor scroll. The DECLARE statement for the static scrollable cursor would have been previously processed. The executive 6 then calls (at block 202) the parser compiler 8 and optimizer 10 to parse and optimize the OPEN statement. After the OPEN statement is parsed and optimized, the executive 6 calls (at block 204) the structure generator 12 to construct an INSERT command from the SELECT statement in the previously compiled and executed DECLARE statement to populate the rows of the result table 50 with the qualifying rows of the base table 60 (**Chen** Paragraph 0051). **"wherein the cursor is used by the function to access values from rows that are returned from a select statement"** as the declaration of the cursor would provide a SELECT statement specifying columns of

the database table 60 and a WHERE clause including one or more predicates to qualify rows of the database table 60. The data manager 16 would return to the cursor the selected columns in the select list from rows that satisfy the WHERE statement (Chen Paragraph 0032).

Therefore, Chen teaches a cursor which is used to access values from columns and rows of a database specified by the select and where statements.

The combination of Chen's cursor used for accessing values from the rows combined with the Agrawal's function used to count and generate frequent itemsets teaches the argued limitation as a whole.

Further Appellant argues that the combination of **Agrawal and Chen** is Improper.

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case, Agrawal teaches select statements for counting and finding frequent Itemsets and Chen teaches select statements providing the declaration of the cursor specifying the columns or rows of a database table. Therefore, It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine

the teaching of the cited references because **Chen's** teachings would have allowed **Agrawal** to provide high concurrency for the rows in the base table when cursors are used by obtaining a lock on the rows in the base table for the duration of the cursor operation.

Appellant's arguments directed towards the rejections of dependent claims 2, 4-7, 12-15, 17-20, and 25-28 reiterate deficiencies Appellant made in the rejection of the independent claim 1 and do not address any new points. Therefore examiner submits that if the rejection of the independent claim is deemed proper, the rejection of claims 2, 4-7, 12-15, 17-20, and 25-28 should also be upheld.

B. §103(a) rejection of claims 3, 9-11, 16, and 22-24 over Agrawal in view of Chen further view of Bayardo.

Appellant argues that **Agrawal, Chen and Bayardo** do not teach or suggest that **"the additional criteria specify a minimum length; and the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results exclude all item sets that include fewer items than the minimum length specified by the additional criteria"** as recited in dependent claim 9.

In response to the preceding arguments examiner respectfully submits that **Agrawal** teaches **"the additional criteria specify a minimum length; and the step of performing the frequent itemset operation includes performing a frequent itemset**

operation whose results exclude all item sets that include fewer items than the minimum length specified by the additional criteria" as combinations of items whose support is greater than minimum support. Call those combinations frequent itemsets (**Agrawal** Col 5, Lines 21-23). The function collects the tids of all tuples of T with the same item in memory and outputs a (item, tid-list) tuple for items that meet the minimum support criterion (**Agrawal** Col 12, Lines 52-55).

Agrawal further teaches the function collects the tids of all tuples of T with the same item in memory and outputs a (item, tid-list) tuple for items that meet the minimum support criterion. The tid-lists are represented as BLOBs and stored in a new TidTable with attributes (item, tid-list) (**Agrawal** Col 11, Lines 49-56). *These lines teach performing frequent itemset operation on combination of items whose support is greater than a minimum support, which will exclude all the combination of items that are below the minimum support.*

Agrawal teaches the elements of argued limitation as noted above but does not explicitly teaches **"a minimum length."**

However, **Bayardo** teaches **"a minimum length"** as Max-Miner usually performs less database passes than this bound in practice when the longest frequent itemsets are more than 10 in length (**Bayardo** Col 9, Lines 57-60). *Examiner interprets the length of 10 as the minimum length.*

Therefore the combination of Bayardo which teaches itemsets of length 10 or more combined with teaching of Agrawal teaches the limitation as a whole.

Appellant argues that **Agrawal, Chen and Bayardo** do not teach or suggest that **"the additional criteria specify a maximum length; and the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results exclude all item sets that include more items than the maximum length specified by the additional criteria"** as recited in dependent claim 10.

In response to the preceding arguments examiner respectfully submits that **Agrawal** teaches **"the additional criteria specify a maximum length; and the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results exclude all item sets that include more items than the maximum length specified by the additional criteria"** as F consists of k+2 attributes (item.sub.1, . . . , item.sub.k, support, len), where k is the size of the largest frequent itemset and len is the length of the itemset (**Agrawal** Col 8, Lines 4-6).

Agrawal further teaches in particular, it is not practical to assume that all items in a transaction appear as different columns of a single tuple because often the number of items per transaction can be more than the maximum number of columns that the database supports. For instance, for one of our real-life datasets the maximum number of items per transaction is 872 and for another it is 700 (**Agrawal** Col 5, Lines 56-60). *These lines teach performing frequent itemset operation on combination of items whose have maximum number of items with size k and length of itemset.*

Agrawal teaches the elements of claim 10 as noted above but does not explicitly teaches **"a maximum length."**

However, **Bayardo** discloses **"a maximum length"** as the most part, frequent-pattern mining methods have been developed to operate on databases in which the longest frequent patterns are relatively short, e.g., those with less than 10 items (**Bayardo** Col 1, Lines 22-26). Examiner interprets the length of 10 as the maximum length.

Therefore the combination of Bayardo which teaches itemsets of length 10 or less combined with teaching of Agrawal teaches the limitation as a whole.

Appellant argues that **Agrawal, Chen and Bayardo** do not teach or suggest that **"the additional criteria specify a set of one or more included items; and the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results exclude all itemsets that do not include all items in said set of one or more included items"** as recited in dependent claim 11.

In response to the preceding arguments examiner respectfully submits that **Agrawal** teaches **"the additional criteria specify a set of one or more included items; and the step of performing the frequent itemset operation includes performing a frequent itemset operation whose results exclude all itemsets that do not include all items in said set of one or more included items"** as the frequent (n+2)-itemsets are determined using cascaded subqueries by: a) selecting distinct first items in the candidate itemsets using a subquery. In generating rules from the union of the frequent itemsets, all items from the frequent itemsets are first put into a table F. These candidate rules are joined with the table F, and filtered to remove those that do

not meet a confidence criteria (**Agrawal** Col 3, Lines 2-16). *Examiner interprets the confidence criteria as determining the set of one or more included items.*

Appellant's arguments directed towards the rejections of dependent claims 3, 16, and 22-24 reiterate deficiencies Appellant made in the rejection of the independent claim 1 and dependent claims 9-11 and do not address any new points. Therefore examiner submits that if the rejection of the independent claim 1 and dependent claims 9-11 is deemed proper, the rejection of claims 3, 16, and 22-24 should also be upheld.

C. §103(a) rejection of claims 29 and 30 over Agrawal in view of Chen further view of Tamura.

Appellant argues that **Agrawal, Chen and Tamura** do not teach or suggest that **"the additional criteria specify a set of one or more excluded items and the step of performing the frequent itemset operation includes performing a frequent itemset operations whose results exclude all itemsets that include all items in said set of one more excluded items"** as recited in dependent claim 29.

In response to the preceding arguments examiner respectfully submits that **Tamura** discloses **"the additional criteria specify a set of one or more excluded items and the step of performing the frequent itemset operation includes performing a frequent itemset operations whose results exclude all itemsets that include all items in said set of one more excluded items"** as an itemset including a

combination of items, which is not in the frequent (k-1)-itemset, is excluded from the candidate k-itemset (**Tamura** Paragraph 0021).

Therefore these lines teach performing frequent itemset operation, which excludes items from the candidate k-itemset which are not in the frequent k-1 itemset. Examiner interprets the items which are not in the frequent (k-1) itemset as one or more excluded items.

Appellant's arguments directed towards the rejections of dependent claim 30 reiterate deficiencies Appellant made in the rejection of the independent claim 1 and dependent claim 29 and do not address any new points. Therefore examiner submits that if the rejection of the independent claim 1 and dependent claim 29 is deemed proper, the rejection of claim 30 should also be upheld.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Usmaan Saeed

Examiner, Art Unit: 2166

/Usmaan Saeed/

Examiner, Art Unit 2166

Conferees:

/Hosain T Alam/

Supervisory Patent Examiner, Art Unit 2166

/Mohammad Ali/

Supervisory Patent Examiner, Art Unit 2158